AMENDMENTS TO THE CLAIMS:

Please amend claims 1 and 9 as follows:

1. (Currently Amended) A method for testing visibility of graphics primitives, said method comprising the steps of:

computing the geometry of graphics primitives;

testing the visibility of the computed primitives in a first visibility test;

storing occlusion data of the visible primitives for a next comparison based on said first visibility test;

computing the occlusion data for each visible primitive;

collecting said primitives to an occlusion data buffer, wherein said occlusion data buffer stores vertex data associated with the primitives, which are formed by at least three vertices;

testing the visibility of the collected primitives in a second visibility test with said computed occlusion data; and

rasterising visible primitives of the second visibility test.

2. (Previously Presented) The method according to claim l, further comprising the steps of discarding hidden primitives of the first visibility test.

3. (Previously Presented) The method according to claim 1, further comprising the step of storing Z values to an occlusion fusion cache while computing occlusion.

4. (Previously Presented) The method according to claim 1, after said first visibility test, collecting occlusion data of the visible primitives belonging to the frame to be rendered to the occlusion data buffer.

5. (Previously Presented) The method according to claim 1, wherein after said visibility test, collecting a predefined amount of occlusion data of the primitives to the occlusion data buffer.

6. (Previously Presented) The method according to claim l, further comprising the step of compressing the occlusion data buffer.

7. (Previously Presented) The method according to claim 1, further comprising the step of testing visibility of an object before the geometry processor by a bounding volume method.

8. (Previously Presented) The method according to claim 1, further comprising the step of testing the visibility of the primitive in the first and the second visibility test with low resolution Z-buffer.

9. (Currently Amended) A system for testing visibility of graphics primitives, said system comprising:

a Geometry processor;

a Z-buffer component;

a first visibility test module;

an occlusion fusion unit;

a pixel processing means;

an occlusion data buffer, wherein said occlusion data buffer stores vertex data associated with primitives, which are formed by at least three vertices; and

a second visibility test module.

10. (Previously Presented) The system according to claim 9, wherein the first visibility test is arranged to discard hidden primitives.

11. (Previously Presented) The system according to claim 9, wherein the occlusion data buffer is arranged to collect occlusion data of the primitives belonging to the frame to be rendered.

12. (Previously Presented) The system according to claim 9, wherein the occlusion data buffer is arranged to collect a predefined amount of occlusion data of the

primitives.

13. (Previously Presented) The system according to claim 9, further comprising means for compressing and decompressing the occlusion data buffer.

14. (Previously Presented) The system according to claim 9, further comprising means for bounding volume testing.

15. (Previously Presented) The system according to claim 9, further comprising an occlusion fusion cache.

16. (Previously Presented) The system according to claim 15, wherein the Z-buffer connected to the first visibility test module is a low resolution Z-buffer.

17. (Previously Presented) The system according to claim 16, further comprising a high resolution Z-buffer connected to said second visibility test.

18. (Previously Presented) The system according to claim 16, wherein values stored to the low resolution Z-buffer are calculated in the occlusion fusion cache.